

IN THE CLAIMS

A complete listing of all claims in the application is included below. As indicated below, please cancel claims 1-21 without prejudice, and add new claims 22-36.

1. (canceled)
2. (canceled)
3. (canceled)
4. (canceled)
5. (canceled)
6. (canceled)
7. (canceled)
8. (canceled)
9. (canceled)
10. (canceled)
11. (canceled)
12. (canceled)
13. (canceled)
14. (canceled)
15. (canceled)
16. (canceled)
17. (canceled)
18. (canceled)
19. (canceled)

20. (canceled)

21. (canceled)

22. (new) A method for processing graphical data, the method comprising:

mapping a convention-defined vertex parameter with an application-defined identifier, wherein the application-defined identifier is definable by an application-defined vertex program and wherein the convention-defined vertex parameter is associated with vertex data;

retrieving the vertex data associated with the convention-defined vertex parameter by calling the application-defined identifier; and

generating a graphical output using the retrieved vertex data.

23. (new) The method of claim 22, wherein the convention-defined vertex parameter comprises an OpenGL defined parameter.

24. (new) The method of claim 22, wherein the convention-defined vertex parameter comprises a D3D defined parameter.

25. (new) The method of claim 22, further comprising:

retrieving additional vertex data by calling the convention-defined vertex parameter.

26. (new) The method of claim 22, wherein the application-defined identifier corresponds to a vertex attribute register.

27. (new) A method for graphics processing, the method comprising:
- receiving a convention-defined vertex parameter;
 - determining an application-defined vertex-attribute identifier associated with the convention-defined vertex parameter; and
 - passing the determined application-defined vertex-attribute identifier to an application-programmable vertex program.
28. (new) The method of claim 27, further comprising:
- retrieving the vertex data associated with the convention-defined vertex parameter by calling the application-defined vertex-attribute identifier; and
 - generating a graphical output using the retrieved vertex data.
29. (new) A method for integrating application-programmable vertex processing with conventional vertex processing, the method comprising:
- mapping an application-defined identifier to a convention-defined vertex parameter, wherein the application-defined identifier is definable by an application-defined vertex program and wherein the application-defined identifier is associated with vertex data;
 - retrieving the vertex data associated with the application-defined identifier by calling the convention-defined vertex parameter; and
 - generating a graphical output using the retrieved vertex data.

B1
cont

30. (new) The method of claim 29, wherein the convention-defined vertex parameter comprises an OpenGL defined parameter.

31. (new) The method of claim 29, wherein the convention-defined vertex parameter comprises a D3D defined parameter.

32. (new) The method of claim 29, further comprising:
retrieving additional vertex data associated with the convention-defined vertex parameter by calling the application-defined identifier.

33. (new) The method of claim 29, wherein the application-defined identifier corresponds to a vertex attribute register.

34. (new) A method for processing graphical data, the method comprising:
receiving an application-defined vertex-attribute identifier from a vertex program;
identifying a convention-defined vertex parameter associated with the application-defined vertex-attribute identifier; and
passing the identified convention-defined vertex parameter to a vertex transformer.

35. (new) The method of claim 34, wherein passing the identified convention-defined vertex parameter comprises:

passing the identified convention-defined vertex parameter to a GL vertex transformer.

13B 36. (new) The method of claim 34, further comprising:

retrieving the vertex data associated with the application-defined vertex-attribute identifier by calling the convention-defined vertex parameter; and
generating a graphical output using the retrieved vertex data.
